AMENDMENTS TO THE SPECIFICATION

In the last paragraph on page 7, which continues onto page 8:

As described above, a VMM presents to other software (i.e., "guest" software) the abstraction of one or more virtual machines (VMs). Figure 1 shows two VMs, 102 and 114. The guest software of each VM includes a guest OS such as a guest OS 104 or 106 and various guest software applications 108-110. Each of the guest OSs 104 and 106 expects to control access to physical resources (e.g., processor registers, memory and memory-mapped I/O devices) within the hardware platform on which the guest OS 104 or 106 is running and to perform other functions. However, in a virtual-machine environment, the VMM 112 should be able to have ultimate control over the physical resources to provide proper operation of VMs 102 and 114 ~~112~~ and protection from and between VMs 102 and 114. The VMM 112 achieves this goal by intercepting all accesses of the guest OSs 104 and 106 to the computer's physical resources. Various techniques may be used to enable the VMM 112 to intercept the above accesses. One of such techniques is a guest-deprivileging technique which forces all guest software to run at a hardware privilege level that does not allow that software access to certain hardware resources. As a result, whenever the guest OS 104 or 106 attempts to access any of these hardware resources, it "traps" to the VMM 112, i.e., the VMM 112 receives control over an operation initiated by the guest OS if this operation involves accessing such hardware resources.

In the first full paragraph on page 8:

Figure 2 illustrates a prior art embodiment of the operation of a VMM that supports guest deprivileging. As described above, guest deprivileging forces a guest OS to execute in a less than privileged mode of execution. For IA-32 microprocessors, the nature of page-based protection is such that all guest software runs at the least privileged level (i.e., ring 3). That is, a guest OS 206 and guest applications 204 run at the same privilege level. As a result, the guest OS 206 may not be able to protect itself from the

guest applications 204 ~~206~~, thereby possibly compromising the integrity of the guest OS 206. This problem is known as ring compression.